

COMMUNICATIONS CUT-OFF BY AUTOMATED DEFENSES FEEDING IPTABLES

Reverse-DNS identity spoofing in Apache / Fail2Ban

Production-validated | iptables / nftables | Critical-service scenarios

RESEARCH AUTHOR

Hugo Vázquez Caramés

ORGANIZATION

Pentest®

OFFICIAL WEBSITE

https://www.pentest.es

A legitimate peer can lose connectivity because a different host triggered a ban.

Power-plant artwork illustrates sector relevance; it is not the tested facility.

01 / EXECUTIVE ASSESSMENT

A defensive tool became the denial-of-service mechanism

The discovery exposes a dangerous trust-boundary failure: a security system accepts a network identity supplied indirectly by reverse DNS, treats it as the source of suspicious traffic, and authorizes a firewall to deny access to that identity. The address blocked is not necessarily the address that made the connection.

Impact: selective denial of service against unrelated systems

When the affected IP belongs to an operator, gateway, service dependency, payment partner or industrial communication endpoint, the defender's own automated protection becomes a targeted availability weapon. The consequences depend on which services the enforcement rule covers.

What is established — and what is not

Evidence tier	Finding	Assessment
CONFIRMED	A live production Apache + Fail2Ban deployment attributed suspicious HTTP activity to an unrelated network address and created a real temporary firewall ban.	End-to-end reproduction
CODE-PATH SUPPORTED	A second form, using a reverse-DNS name that visually resembles a numerical IP address, traverses relevant parsing paths in reviewed configurations.	Not separately reproduced
NOT ESTABLISHED	Prevalence across default installations; compromise of energy, railway or financial systems; exploitation of all products in the comparison matrix.	Scenario analysis only

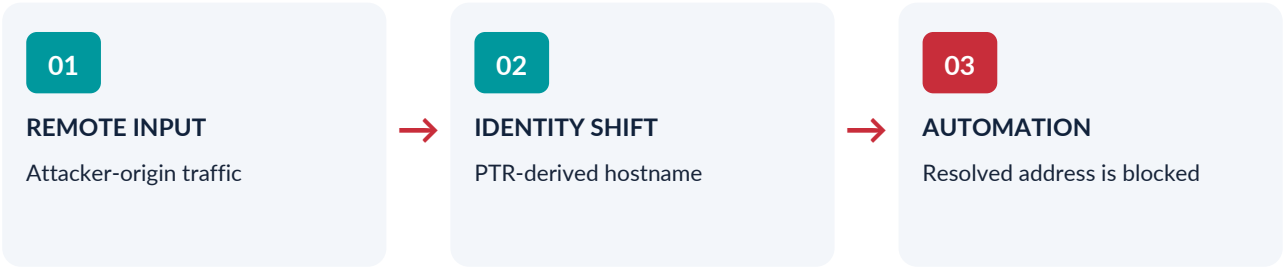
Risk classification: **HIGH for affected deployments with consequential enforcement scope**; potentially critical when automatic blocking sits on an essential communications path. This is a configuration-dependent weakness, not a claim that every Fail2Ban installation is vulnerable.

The weakness is especially insidious because the audit trail can look internally consistent: the hostname in the web log resolves to the IP in the ban record. The missing fact is the original network-layer source address.

02 / FAILURE MODE

From untrusted name to trusted firewall action

Reverse DNS (PTR) is descriptive metadata. It is **not proof of the identity of the host that opened a TCP connection**. Yet many security workflows are optimized for consuming already-formatted logs rather than retaining authoritative network-layer provenance.



How exploitation works — at a conceptual level

An adversary capable of influencing the reverse-DNS name associated with a connection presents an apparently legitimate hostname to a server that performs reverse lookups. The web server may record that hostname rather than the original numeric source. A downstream log-based blocking engine then associates a suspicious event with the DNS-resolved address of that hostname. If an enforcement threshold is reached, the rule is applied against the **unrelated address**.

The attacker does not need to control the targeted organization's DNS, operating system or firewall. The critical error occurs when defender-controlled automation promotes a lower-trust naming assertion into an enforcement identity. The exact preconditions vary by logging format, resolver behavior, filter, and action.

Conditions required for the reproduced Fail2Ban path

- A server records reverse-DNS hostnames in a field later interpreted as the client identity.
- An enabled log filter accepts hostnames rather than requiring original numeric connection IPs.
- The blocking engine performs forward resolution and trusts the resulting address without comparing it to the source connection.
- An active enforcement action applies to the services reached by the selected address.

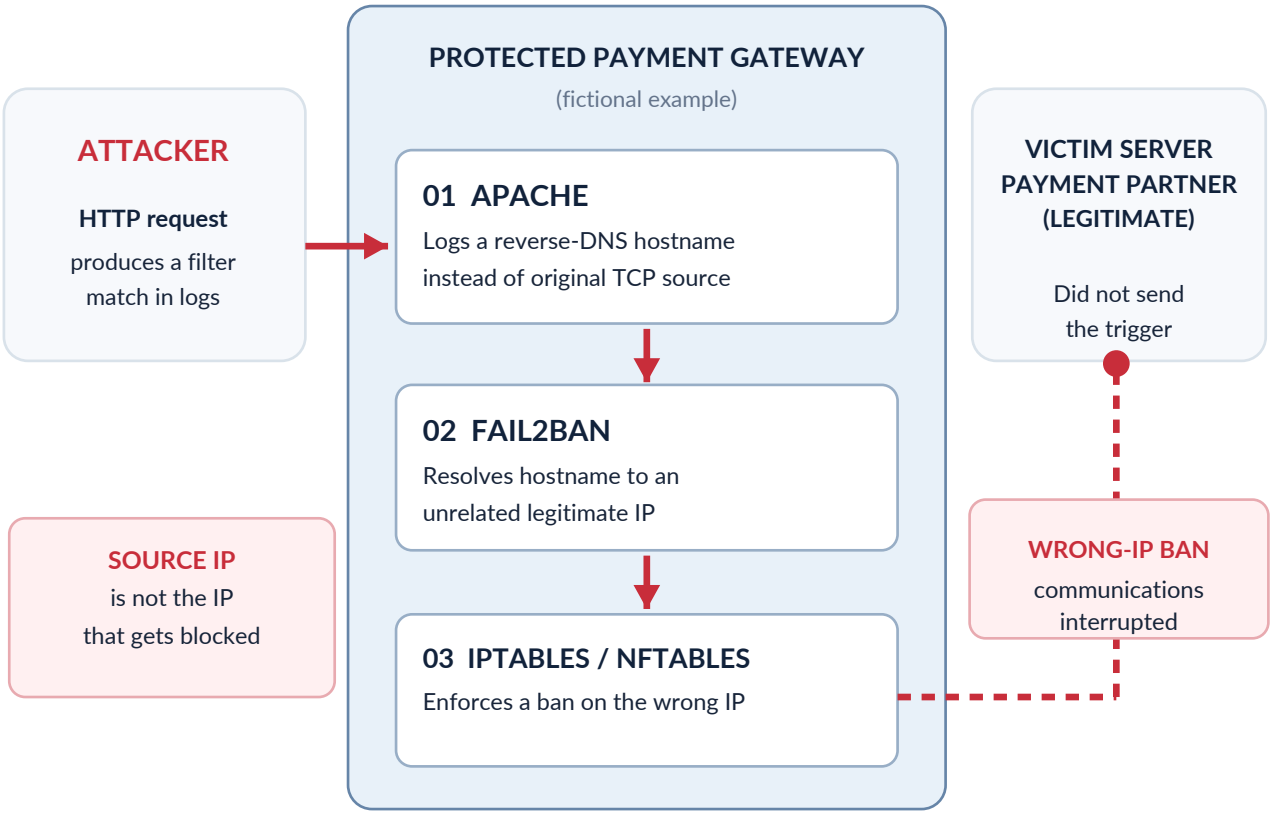
This is identity laundering through telemetry.

An untrusted DNS description crosses two boundaries — into a log identity, then into an authoritative firewall decision — without binding the final IP back to the original connection.

03 / ATTACK FLOW - FICTIONAL PAYMENT GATEWAY

One forged identity. One defensive ban. One communications outage.

This schematic shows the **mechanism** using a fictional payment-processing system. The attacker originates the HTTP event; the protected gateway evaluates its own logs; a legitimate payment partner that did not send the event becomes the ban target. The diagram is not the architecture of the production validation.



Trust boundary crossed: PTR metadata -> log identity -> firewall decision

Simplified impact model; not a disclosure of the actual production topology.

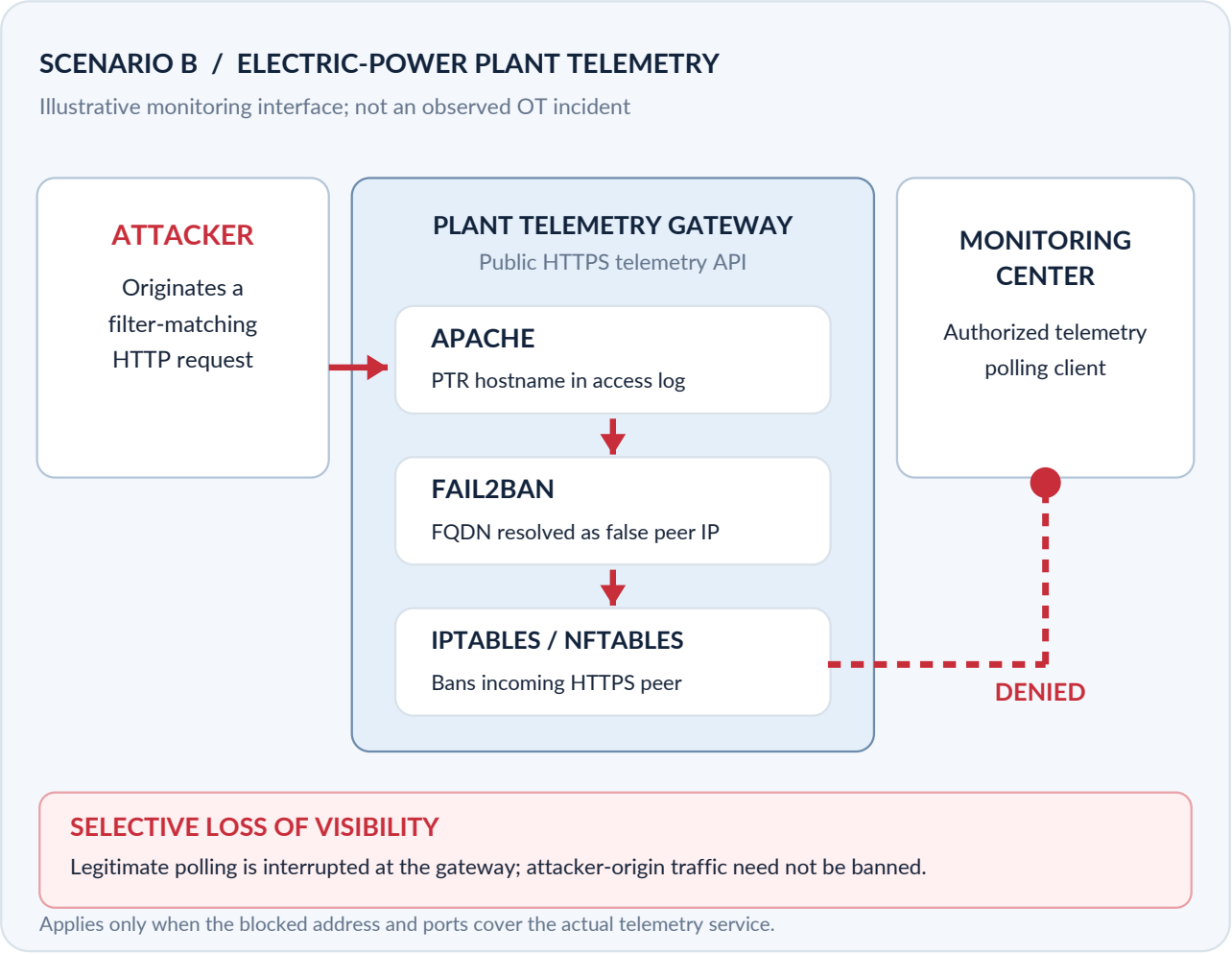
This is the distinctive threat: defensive enforcement against an innocent dependency.

The protected gateway can remain healthy for most users even as a selected processor, acquirer or callback peer is denied service. The availability loss depends on which ports and destinations the rule actually protects.

04 / ATTACK FLOW - ELECTRIC-POWER TELEMETRY

A monitoring center disconnected by its own plant gateway

A fictional generating station exposes a protected **HTTPS telemetry interface** to an authorized off-site monitoring center. An adversary-origin HTTP event causes an unsafe reverse-DNS identity to travel through the gateway log and Fail2Ban. The enforcement layer then blocks the **monitoring center**, not the sender of the suspicious request.



Risk: monitoring continuity can fail while the defended server remains available to other peers.

If the banned source is the only authorized telemetry poller, external visibility may be interrupted. This diagram does not imply compromise of plant control systems, loss of physical generation, or an actual utility outage.

Scope: *the example requires a telemetry API reachable through the filtered ports, a matching hostname-based log pipeline, a DNS-to-IP decision and enforcement against the legitimate poller. It is not an assertion about conventional isolated control networks.*

05 / PRODUCTION VALIDATION

Observed in a real, Internet-facing production environment

On **3 November 2026**, the research reproduced incorrect-source blocking on a functioning, publicly reachable production web service using Apache and Fail2Ban 1.0.2. The service was not an isolated emulator. The production environment was controlled by the researchers; participating addresses were authorized for testing.

Evidence preserved from the observed sequence

Stage	Direct observation	What it establishes
1 / Web request	A request reached the live web server from an originating IP.	Actual network event, not fabricated log input.
2 / Logging	The HTTP access log recorded a PTR-derived hostname in place of the connecting IP.	Source identity in telemetry diverged from transport reality.
3 / Detection	The stock Fail2Ban filter matched the request; the active jail resolved the recorded hostname.	Name interpretation influenced security decision.
4 / Enforcement	Fail2Ban recorded a **Found** event and a **Ban** for a different, researcher-controlled address.	The wrong IP entered the active enforcement workflow.
5 / Recovery	An **Unban** event followed the temporary ban; manual access tests verified selective connectivity impact.	The event was a real, reversible firewall action.

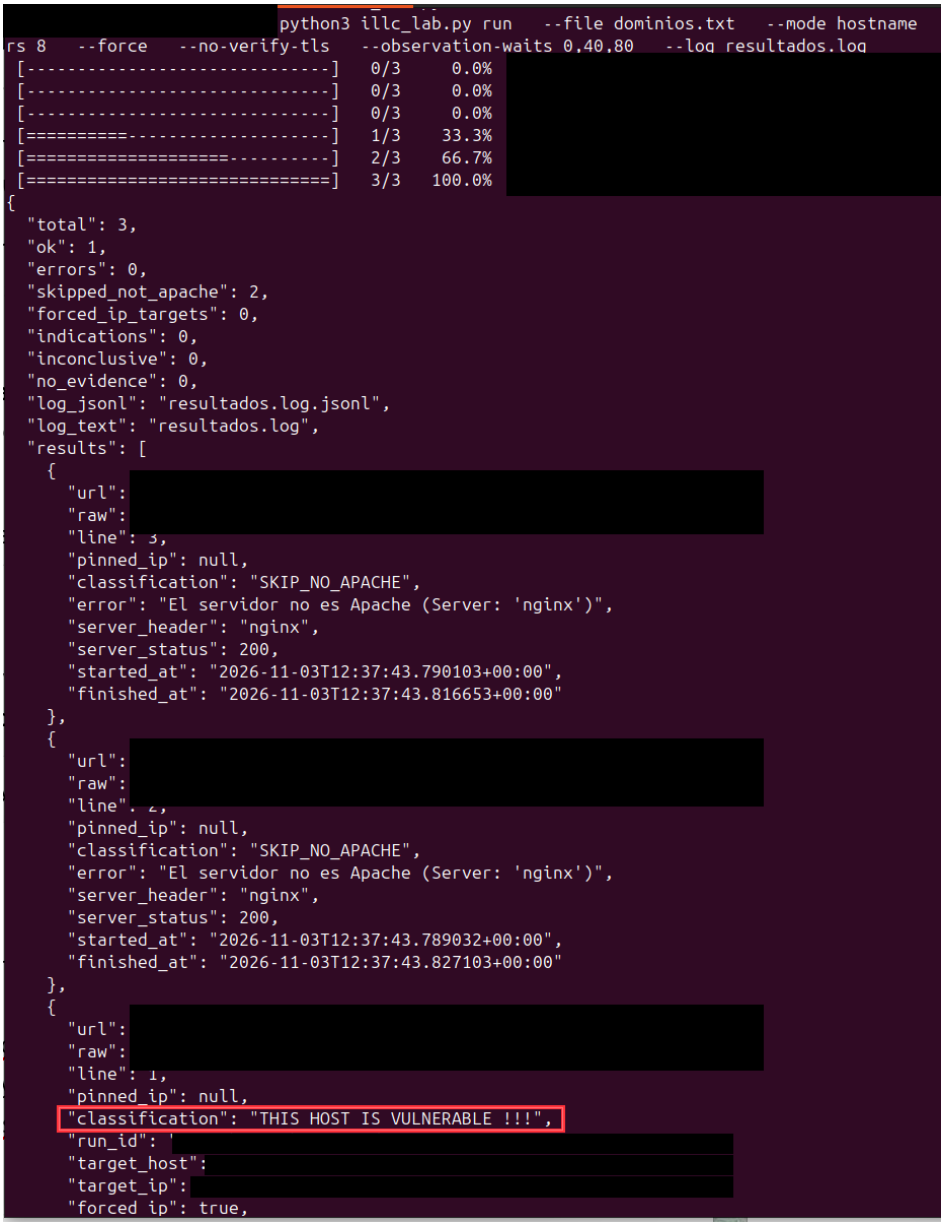
Result: confirmed false-positive ban with real availability consequences.

The IP banned by Fail2Ban did not initiate the triggering HTTP connection. This is the central exploitability finding — demonstrated on the live deployment, not merely inferred from source code.

06 / REDACTED EVIDENCE

Production result, without exposing the test environment

Below is a **redacted original terminal capture** from the research workflow. All displayed network addresses, hostnames, domains and workstation identifiers have been irreversibly covered. The terminal visibly reports **THIS HOST IS VULNERABLE !!!** for one assessed host and skipped two unrelated NGINX hosts. The classification is highlighted in red.



Evidence discipline: the automated classification in this screenshot is not the sole proof. The production finding was separately corroborated by Apache access records, Fail2Ban *Found/Ban/Unban* events and manual connectivity checks from independently controlled endpoints. Identifiers are intentionally not recoverable from this image.

07 / HISTORICAL PRECEDENT

ILLC: a two-decade-old warning revisited

The research was inspired by **Inverse Lookup Log Corruption (ILLC)**, an early security problem described in connection with reverse DNS and web-server logging. Historic identifiers **CVE-2003-1580** and **CVE-2003-1581** describe related but distinct effects; these are historical references, **not** new CVE assignments for Fail2Ban. [4, 5]

Variant A — hostname-based attribution

A reverse-DNS response supplies an ordinary hostname. If a log-based blocking tool later resolves that name to a different address, the final action is detached from the real client. **This is the variant experimentally confirmed in production** with Apache and Fail2Ban.

Variant B — numeric-looking PTR identity

A reverse-DNS hostname can be crafted to *look* like an IPv4 address. For example, an illustrative connection originating from **198.51.100.23** could be represented in a reverse-resolved log by the hostname **192.0.2.44.**, subsequently appearing as **192.0.2.44**. The addresses are documentation-only examples.

Second variant: supported by code review; not yet proven end-to-end.

The reviewed Apache log-format and Fail2Ban address-matching paths indicate a credible route for numeric-looking PTR names to be treated as literal IP identities. That result is an audit-derived exploitability assessment, not a claim of a completed firewall-ban demonstration. Resolver behavior and deployment configuration remain decisive.

CVE-2003-1580 specifically concerns ambiguous dotted-quad reverse-DNS names; CVE-2003-1581 concerns arbitrary log-text injection in a historical Apache context. The newly demonstrated effect is **downstream authorization of a wrong-address ban**, which makes the operational consequence more severe than mere log inaccuracy. [4, 5]

Default Apache behavior is important: **HostnameLookups Off** avoids this particular ordinary reverse-lookup path. **Double** requires forward resolution to include the originating address and can reject an inconsistent conventional hostname. [3]

08 / CVSS AND SCALABILITY

A 2.6 score cannot tell an operator what will go dark

NVD records **CVE-2003-1581**, an Apache 2.0.44 log-injection variant of ILLC, with a legacy **CVSS v2.0 base score of 2.6 / 10 (LOW)**, vector **AV:N/AC:H/Au:N/C:N/I:P/A:N**. Crucially, that historical metric assumes **no availability impact** for the vulnerability it describes. [5]

NVD CVSS v2 base score: 2.6 - yet defensive automation can disconnect legitimate peers.

The historic 2.6 score is not a severity rating for the newly observed Fail2Ban downstream behavior. But it is a stark warning against treating a low historical base score as a complete operational-risk assessment where the same trust-boundary failure is connected to automated firewall enforcement.

Risk position: for production systems that automatically turn a DNS-derived identity into a network deny rule, severity must be evaluated against actual connectivity consequences. In an OT control gateway or financial service perimeter, selective loss of a single critical peer may warrant HIGH or CRITICAL organizational treatment even if the ancestor CVE was assigned LOW under an older scoring framework. A scoring system is a prioritization input, not a substitute for consequence engineering.

Can the mechanism scale beyond one peer or one protected host?

Dimension	Technical risk	Evidence boundary
Multiple protected hosts	The same failure mode may exist on several services whose logging and enforcement share the unsafe identity chain; each target must be assessed independently.	**Possible**; simultaneous exploitation of several production hosts not demonstrated.
Multiple victim IPs	Separate poisoned identities and repeated matching events can potentially produce more than one wrong-address entry in a ban set.	**Plausible extension**; the verified experiment blocked one unrelated IP.
Entire ranges (CIDR)	A whole prefix can be disrupted only if an action, policy or orchestration layer permits prefix-level bans or expands multiple addresses.	**Conditional scenario only**; a stock single-IP Fail2Ban ban does not imply automatic /24 or /16 blocking.

Bottom line: even one wrongly banned upstream peer can be an availability incident. A broader cascade would amplify the consequence, but is not being presented as a completed production test.

09 / EXPOSURE MATRIX

Which products and configurations deserve scrutiny?

The question is not whether a product parses text; it is whether a supplied identity can travel from a log into an enforceable IP action. The matrix separates confirmed exploitation from potential exposure.

Technology	Conditions / trust boundary	Evidence status
Apache HTTP Server	Reverse-lookup-enabled hostname logging is the upstream input. Numeric client-IP logging and double verification change the outcome.	**Production confirmed** as source of misleading log identity. [3]
Fail2Ban	Hostname-capable filters (for example, rules accepting hostnames), DNS resolution of log identities, enabled jail and blocking action. Not the default of every filter or jail.	**Production confirmed** for one official-filter configuration; other combinations vary. [1, 2]
SSHGuard	A relevant log source is supplied and a hostname-bearing event is parsed; reviewed code resolves hostnames to IP addresses.	**Code-path exposure**; integrated false ban not reproduced. [13]
CrowdSec	Apache parser accepts host-or-IP syntax, but downstream IP typing and decision logic may reject FQDNs.	**Parser exposure only**; wrong-IP blocking not established. [12]
Wazuh	Certain web-log decoders populate a source-IP field from log text; active response has further validation requirements.	**Review candidate**; no integrated ban demonstrated. [14]
NGINX	Default access logging uses numeric remote address; exposure requires nondefault hostname enrichment or later transformation.	**Not shown** in default logging path. [15]
pfSense + SSHGuard	Native protection typically monitors authentication events, not arbitrary Apache access logs.	**No default-route finding**; separate trust chains must be assessed.

Key caution: *mention of a product in this matrix does not constitute an advisory against all versions or all installations. No deployment survey or prevalence estimate has been completed.*

10 / CODE AUDIT - APACHE HTTP SERVER

The first identity substitution: %h is not the TCP peer

The upstream trust boundary is located inside Apache logging. In Apache 2.4, **modules/loggers/mod_log_config.c** implements distinct emitters for hostnames and addresses. When reverse lookups are enabled, a hostname may replace the connection identity displayed in a log. The downstream defender receives text, not proof of provenance. [3, 16]

Named functions, variables and security interpretation

Code symbol / location	Code-level behavior	Audit implication
log_remote_host() mod_log_config.c	Delegates to **ap_get_useragent_host(..., REMOTE_NAME, ...)** (or connection variant) and escapes the log item.	Produces **%h**: a remote hostname, not a guaranteed numeric TCP address.
log_remote_address() mod_log_config.c	Returns **r->useragent_ip**; the connection-specific branch uses **r->connection->client_ip**.	Distinguishes **%a** from **%{c}a**; proxy modules can affect the apparent user-agent address.
HostnameLookups Apache core	Off (default) avoids routine PTR substitutions; On performs reverse name lookup; Double adds a forward verification step.	A forward check can reject a mismatched PTR/A pair but is not a replacement for trusted transport identity.

The exact source field is the security boundary.

If a ban engine consumes %h as though it were %a or an authenticated proxy-origin address, an externally influenced DNS name becomes an enforcement identity. This finding was confirmed for the tested Apache logging configuration; it is not Apache remote code execution.

Proxy and reverse-proxy caveat

With **mod_remoteip**, the apparent client address can be rewritten from configured proxy headers. Consequently, a remediation rule that blindly substitutes %a for %h may still fail if proxies or headers are not trusted correctly. For a direct connection, **%{c}a** is the underlying peer; behind a trusted reverse proxy, the application-level origin must be obtained from a separately authenticated boundary. [16, 3]

11 / CODE AUDIT - FAIL2BAN PARSER

The second substitution: DNS text becomes a ban identity

The **Fail2Ban 1.0.2** code exposes the exact transition from a matched log field to a typed address. The relevant paths are **config/filter.d/php-url-fopen.conf**, **fail2ban/server/failregex.py**, **fail2ban/server/filter.py** and **fail2ban/server/ipdns.py**. This is the version family validated in production. [2, 17, 18, 19]

What each code component does

File / method	Source-level decision	Result
filter.d/php-url-fopen.conf	The packaged failregex places **<HOST>** in the leading identity position of an Apache log line.	The security identity is inherited from the log, not from the original socket.
failregex.py / getFailID()	Regex captures can carry **ip4**, **ip6** or **dns** identifiers.	Hostnames and literal addresses traverse related but distinguishable branches.
filter.py / findFailure()	Reads **fail["ip4"]**, **fail["ip6"]** and **fail["dns"]**; non-raw values enter DNS conversion.	A hostname recorded by Apache can become an IP used in the violation record.
ipdns.py / textToIp()	Looks for an address-like token, then calls **dnsToIp()** when **usedns** allows resolution.	The program trusts resolved identity sufficiently to proceed to blocking logic.
ipdns.py / dnsToIp()	Calls **socket.getaddrinfo()** for IPv4 and IPv6 and gathers the results.	The answer is not reconciled against the original Apache TCP peer.

The verified configuration was sensitive to this path.

The live jail reported usedns=yes; the filter matched; the recorded DNS-derived identifier resolved to a different controlled IP; and Fail2Ban logged Found/Ban for that IP. The code review explains the observed chain, but is not being generalized to every version or filter.

Numeric-looking PTR variant: source branches for literal IPv4 strings and DNS names make a dotted-quad-shaped PTR an important separate review case. Handling of trailing dots, normalization and resolver rules must be verified for each deployment; the variant has **not** been validated end-to-end here. [17-19]

12 / CODE AUDIT - FAIL2BAN ACTIONS

How an incorrect identity reaches the kernel firewall

Detection alone is not the outage. The material security consequence appears when a **FailTicket** for the resolved identifier reaches an enabled action. In 1.0.2, **filter.py / processLineAndAdd()** creates the failure ticket and updates the failure manager; the configured ban action applies its selected IP to an enforcement structure. [17, 20]

Important branches and their consequences

Path or configuration	Observed source behavior	Operational consequence
processLineAndAdd()	Creates **FailTicket(ip, unixTime, ...)**, checks ignored addresses and calls **failManager.addFailure()**.	The security state now tracks a reconstructed address as the offender.
findFailure()	For DNS identifiers, uses **DNSUtils.textToIp(...)** and iterates across returned identifiers.	A hostname with multiple eligible DNS answers can generate multiple failure entries.
dnsToIp()	Accumulates addresses in a **set** returned by getaddrinfo; implements a bounded lookup cache.	DNS answer composition and resolution timing may affect which addresses are considered.
nftables.conf	The **actionban** template inserts the selected **<ip>** into a named nftables set; **actionunban** deletes it.	The misidentified source is converted into a real firewall match.
banaction / port	Action type, chain and port selection determine scope; the action can target different backends.	The same attribution bug has different consequences under ports-only versus broad drop rules.

Multiple addresses and network prefixes: exact evidence boundary

The **multi-address path is explicit in the 1.0.2 code**: the DNS helper may return more than one address and the filter constructs a finding for each returned identifier. Whether each is eventually banned depends on jail thresholds, ignore lists, duplicate suppression and action compatibility. **Whole-prefix blocking** is not a demonstrated default behavior: a /24 or /16 impact requires an action capable of prefix enforcement, an external orchestration policy or many individual IP entries. [17, 19, 20]

Consequence: one DNS identity can expand into several candidate failures. The production proof involved one unrelated IP; multi-address enforcement remains a code-supported conditional risk.

13 / SOURCE REVIEW - SSHGUARD & CROWDSEC

Adjacent defensive engines: similar surfaces, different safeguards

Product names in this chapter are **review candidates, not confirmed vulnerable deployments**. The following symbols are from public project code. End-to-end exploitation was validated only for the Apache + Fail2Ban configuration. [12, 13]

SSHGuard: hostname input reaches a resolver

File: src/parser/attack.c. **Function: attack_from_hostname(attack_t *, const char *)**. The implementation attempts **getaddrinfo()** with **AF_INET**, then **AF_INET6**, selects a returned address and stores it in **attack->address**. This is an identifiable hostname-to-address conversion path. However, its exploitability depends on whether a real SSHGuard parser accepts an attacker-controlled name from a chosen log source and whether the corresponding backend enforces that address. No such integrated exploitation has been demonstrated. [13]

CrowdSec: Apache parser and source identity

File: parsers/s01-parse/crowdsecurity/apache2-logs.yaml. A Grok expression can parse Apache-style lines and an optional **target_fqdn** prefix. The static assignments include **evt.Meta.source_ip** (from **evt.Parsed.clientip**) and **evt.Meta.target_fqdn**. These are distinct fields: the latter identifies a target in multi-tenant collection, not automatically the offender. [12]

Question for audit	SSHGuard	CrowdSec
Entry to trust boundary	Hostname from supported authentication/log parser	Client identity as parsed from Apache-style telemetry
Resolver path identified	Yes: attack_from_hostname() uses getaddrinfo().	Parser field assignment seen; no equivalent wrong-address ban proven.
Mandatory additional proof	Accepted log format + attack source + backend firewall result	Typed source-IP validation + scenario selection + decision + bouncer enforcement
Assessment status	Code-path candidate, not validated here	Parser provenance candidate, not validated here

Parsing a hostname is not the same as banning a hostname.

For both projects the complete path must be traced through normalization, source-IP validation, alert decision and firewall action. No public claims of confirmed vulnerability are justified by a parser fragment alone.

14 / SOURCE REVIEW - WAZUH, NGINX & FIREWALLS

Where code review must stop short of a vulnerability claim

Wazuh: decoded identity versus validated active response

File: decoders/0375-web-accesslog_decoders.xml. Named decoder variants include **web-accesslog-domain**, **web-accesslog-ip-ip** and **web-accesslog-ip**. Regex captures are assigned to ordered fields such as **srcip**, **protocol**, **url** and **id**. The XML expresses the extractor, but does not by itself prove that arbitrary hostnames survive IP typing or reach an active-response firewall rule. Check the validation and active-response pipeline before making that claim. [14, 21]

NGINX: why the standard logging path is materially different

The default NGINX **ngx_http_log_module** combined format begins with **$remote_addr**, a client-address variable, rather than an automatic PTR-derived hostname. This explains why a default NGINX log path is not equivalent to Apache **%h** with reverse lookups enabled. Non-default log enrichment, reverse proxies and downstream transformations require independent review. [15]

pfSense / SSHGuard and generic firewall integrations

Packaging an SSHGuard-like protection engine inside an appliance does not mean that it consumes Apache access logs or uses attacker-provided PTR names. For pfSense or a custom firewall pipeline, the review must identify (i) the emitter of the original log, (ii) the parser, (iii) any DNS resolver, (iv) the typed IP decision and (v) the precise kernel rule applied. **We have not validated pfSense as vulnerable**.

Audit-level decision matrix

Source code signal	Why it matters	Evidence needed to call it exploitable
hostname-to-IP conversion	May detach event identity from transport source.	A controlled, real event causes conversion to an unrelated address.
typed srcip / source_ip field	May preserve, normalize or reject bad values.	Proof the false identity survives field validation.
automatic response / bouncer	Connects detection to network enforcement.	Rule contains wrong address; independent connection loses service.
multiple answers / prefixes	May amplify number or scope of blocks.	Action semantics, thresholds and live result verified separately.

Scope limit: None of the above is a substitute for version-specific tests. Modern code may differ from reviewed branches, and enabled defaults are not uniform across installations. Source files are linked in the references.

15 / INDUSTRIAL RELEVANCE

Fail2Ban is already in the OT security conversation

It would be a serious mistake to dismiss this class of failure as irrelevant to operational technology. Fail2Ban appears in national-laboratory research and in commercial industrial Linux and gateway documentation. That is evidence of **industrial relevance and real integration**, not proof that those particular products inherit the reproduced Apache-filter configuration.

Source	What the official document actually says	Significance
Sandia National Laboratories DOE research, 2021	Reviews Fail2Ban in §2.1.1.5; Table 1 (p. 42) lists it among host-based signature approaches for distributed energy resource (DER) environments.	Direct OT/energy consideration. [6]
NREL / DOE Module-OT, 2022	§4.1.2, p. 20 includes “Fail2ban—SSH administration tool” among software requirements for an OT command-and-control security architecture.	Explicit role in an OT research architecture. [7]
Moxa Industrial Linux 4 2026 manual	Intrusion Prevention section, p. 98: states that Fail2Ban is preinstalled in Moxa Industrial Linux; product table also lists Fail2Ban.	Commercial industrial-platform integration. [8]
Moxa ThingsPro 2 2018 release history	v2.5.0 release history, p. 2: adds Fail2Ban protection for SSH login attempts.	Documented industrial gateway feature. [9]
Eurotech ESF Network Threat Manager	Official product documentation describes Fail2Ban-based intrusion prevention for gateway administration and SSH.	Documented commercial integration. [10]

What NIST says — and does not say

NIST SP 800-82 Rev. 3 discusses unsuccessful login attempts in **AC-7 (printed p. 231)** and recognizes that continuous operation limits where lockout controls can be applied. **NIST does not prescribe Fail2Ban**; nor does documentation of a security control prove that OT process traffic is filtered by that product. [11]

16 / CONSEQUENCE MODEL

A selective blockade can be more damaging than a broad outage

Selective denial of service is strategically different from saturating a network. It can isolate one vital participant while leaving the wider site healthy. The defender's service may appear operational to most users, while one partner, operator or automated integration becomes unreachable.

Illustrative high-consequence scenarios — not observed incidents

Sector / dependency	Possible consequence if the vulnerable enforcement boundary covers critical flows
Electric power / DER	An authorized telemetry or remote-operations source loses access to a gateway. Monitoring, visibility or supervisory workflows may be interrupted; impacts depend on network segmentation and fail-safe design.
Railway systems	A maintenance, monitoring or coordination service is selectively disconnected from a protected interface. Operational impact depends on redundancy and whether the connection is safety relevant.
Financial services	A payment processor, acquiring-bank partner, fraud platform or upstream callback source is denied connectivity, degrading transaction acceptance or settlement-related integrations.
Industrial supply chains	Remote vendor support, machine-to-machine reporting or fleet orchestration loses connectivity to a specific address while routine health checks from elsewhere still pass.

The failure mode turns automated defense into an availability liability.

In organizations that enforce network identity at critical boundaries, a corrupted log identity can trigger decisive firewall action. No malware on the destination, credential theft or volume-based flooding is required by the mechanism itself. Whether disruption propagates into physical or financial operations depends on the actual architecture.

Responsible interpretation: our production reproduction involved a public web service and a controlled test address. We have **not** tested power grids, railway operations, payment gateways or other critical infrastructure. These are risk scenarios explaining why validation matters before deployment in high-availability networks.

17 / ACTIONS FOR DEFENDERS

Stop trusting reverse DNS as a blocking identity

This class of issue should be treated as a data-provenance failure. Fixing one filter without verifying the identity pipeline leaves room for equivalent mistakes elsewhere.

1 Preserve the transport-layer source

Use a trusted numeric IP field captured at the connection or authenticated proxy boundary as the only source for network bans. Keep PTR hostnames for display, not authorization.

2 Review hostname-capable filters

Inventory Fail2Ban jails whose log sources include reverse-resolved names and whose filters accept hostname values. Evaluate whether DNS resolution is required at all; review defaults and overrides.

3 Validate proxy and log pipelines

Ensure reverse proxies, syslog enrichment, parsers, collectors and enrichment jobs cannot silently replace origin IPs with externally controlled names.

4 Constrain consequences

Limit bans to necessary services, set proportionate durations, avoid critical allowlisted management/partner flows and require safeguards before banning high-value dependencies.

5 Conduct controlled black-box verification

Test from separately controlled, authorized identities; document actual connection origins, emitted log fields, security decisions and firewall enforcement. Negative tests are not universal immunity certificates.

6 Address OT / financial availability

Treat automatic deny rules in mission-critical communications paths as change-controlled safety/availability decisions with redundancy and rapid rollback.

Immediate operational priority

Organizations should audit any automatic IP blocking workflow that ingests reverse-DNS-derived hostnames. Where an untrusted name can become a firewall target, the system is one false identity away from blocking a legitimate peer.

18 / THE HUMAN FACTOR IN LOGIC VULNERABILITIES

The missing link was not a line of code. It was a human hypothesis.

In 2003, inverse-lookup logging attacks exposed a trust weakness in reverse DNS. In 2026, this research connected that old warning to modern, automated firewall enforcement. The distance between those dates is **23 years**. A historic CVSS v2 base score of **2.6** did not capture the potential operational consequences of the distinct, modern **log-to-ban chain**. [4, 5]

The critical insight came from connecting systems, not inspecting them in isolation.

A PTR record is often viewed as harmless metadata. A log parser appears to be performing its intended function. A firewall correctly executes the action it receives. Only by joining all three behaviors does the selective denial-of-service emerge.

Why human-led correlation matters

- A **human researcher** recognized that an obscure, historically low-rated identity problem could reappear when combined with automated IP blocking.
- The decisive reasoning crossed boundaries between **DNS, web-server logging, parser semantics, network enforcement and service availability**. These components can look individually correct.
- An AI assistant can accelerate source navigation, code comparison and report drafting; however, it does not eliminate the need to frame the right question, test the assumption against a live boundary and challenge plausible-sounding conclusions.
- Our investigation did **not** demonstrate that all AI systems have failed to discover this relationship, nor that no researcher had previously noticed it. Such universal claims cannot be inferred from this case.

Research lesson: human judgment establishes what evidence means

The meaningful breakthrough was not a classifier matching a suspicious request. It was the decision to ask whether the address in an enforcement rule was **the address that had actually made the connection**. That question is deceptively simple—and fundamental. The path from a 2003-era low-impact description to a 2026 production-validated selective communications cut-off illustrates why logical vulnerabilities demand hypothesis-led, systems-level investigation.

AI is an analytical amplifier, not a substitute for a research lead.

Human researchers must choose the threat model, connect seemingly unrelated components, construct falsifiable tests and own the final interpretation. AI-assisted code review is useful; it is not independent proof of exploitability.

19 / REFERENCES AND EVIDENCE NOTES

Primary sources and verification boundaries

References are official vendor, laboratory, standard-body or project sources. Pagination refers to the printed page where reported; web-only items have no page number. Sources establish product behavior or industrial relevance, **not** exploitation of their specific deployed configurations.

[1] Fail2Ban project: jail.conf — usedns semantics and default jail behavior
github.com/fail2ban/fail2ban/blob/master/config/jail.conf

[2] Fail2Ban 1.0.2: official php-url-fopen filter
raw.githubusercontent.com/fail2ban/fail2ban/1.0.2/config/filter.d/php-url-fopen.conf

[3] Apache HTTP Server 2.4: HostnameLookups directive
httpd.apache.org/docs/2.4/en/mod/core.html#hostnamelookups

[4] CVE-2003-1580: numeric-looking reverse-DNS hostnames (NVD / NIST)
nvd.nist.gov/vuln/detail/cve-2003-1580

[5] CVE-2003-1581: historical log injection variant - CVSS v2.0 base score 2.6 (NVD / NIST)
nvd.nist.gov/vuln/detail/cve-2003-1581

[6] Sandia / DOE (2021), Review of Intrusion Detection Methods and Tools for Distributed Energy Resources, §2.1.1.5 and Table 1 (p. 42)
www.osti.gov/servlets/purl/1769265

[7] NREL / DOE (Jan 2022), Module-OT, NREL/TP-5R00-79974, §4.1.2 (p. 20)
docs.nrel.gov/docs/fy22osti/79974.pdf

[8] Moxa (Feb 2026), Industrial Linux 4.x (Debian 13), Manual for Arm-based Computers, “Intrusion Prevention” (p. 98)
www.moxa.com/getmedia/aa29ec12-a5bd-4cd8-9f66-f6ffe615280e/moxa-mil-4.x-debian-13-arm-ba…

[9] Moxa (2018), ThingsPro 2 v2.5.0 Software Release History (p. 2)
www.moxa.com/Moxa/media/PDIM/S100000390/ThingsPro%202_moxa-thingspro-2-software-package-…

[10] Eurotech, ESF Network Threat Manager — Fail2Ban feature
esf.eurotech.com/docs/network-threat-manager

[11] NIST (2023), SP 800-82 Rev. 3, Guide to Operational Technology Security, AC-7 (p. 231)
nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82r3.pdf

19 / CODE AND PRODUCT REFERENCES

Implementation sources and review boundaries

The public code references below identify concrete functions and configuration paths. A source-level finding alone does not establish exploitability of a deployed product.

[12] CrowdSec Hub: Apache log parser (review candidate)
github.com/crowdsecurity/hub/blob/master/parsers/s01-parse/crowdsecurity/apache2-logs....

[13] SSHGuard: hostname-to-address resolution path (source)
github.com/SSHGuard/sshguard/blob/master/src/parser/attack.c

[14] Wazuh: web access log decoders (source)
github.com/wazuh/wazuh-ruleset/blob/master/decoders/0375-web-accesslog_decoders.xml

[15] NGINX official logging documentation (numeric client-IP formats)
docs.nginx.com/nginx/admin-guide/monitoring/logging/

[16] Apache httpd: modules/loggers/mod_log_config.c - log_remote_host() and log_remote_address()
github.com/apache/httpd/blob/trunk/modules/loggers/mod_log_config.c

[17] Fail2Ban 1.0.2: fail2ban/server/filter.py - findFailure() and processLineAndAdd()
github.com/fail2ban/fail2ban/blob/1.0.2/fail2ban/server/filter.py

[18] Fail2Ban 1.0.2: fail2ban/server/failregex.py - getFailID()
github.com/fail2ban/fail2ban/blob/1.0.2/fail2ban/server/failregex.py

[19] Fail2Ban 1.0.2: fail2ban/server/ipdns.py - DNSUtils.textToIp() and dnsToIp()
github.com/fail2ban/fail2ban/blob/1.0.2/fail2ban/server/ipdns.py

[20] Fail2Ban 1.0.2: config/action.d/nftables.conf - actionban/actionunban
github.com/fail2ban/fail2ban/blob/1.0.2/config/action.d/nftables.conf

[21] Wazuh documentation: ruleset decoder field ordering and source-IP typing
documentation.wazuh.com/current/user-manual/ruleset/ruleset-xml-syntax/decoders.html

Disclosure note: The production test, architecture and exact exploitation stimulus have been omitted from this publication to limit operational abuse. The report is intended to support verification, remediation and coordinated disclosure.

20 / RESEARCH METHODOLOGY AND DISCLAIMER

Manual exploit validation. AI-assisted review and drafting.

This report distinguishes the **manually observed security event** from the **assisted technical analysis** used to prepare its broader discussion. That distinction is essential for accurate reliance on the findings.

What was validated manually

The vulnerability tests underlying the confirmed Apache / Fail2Ban finding were conducted and corroborated **manually by the researcher**. The evidence included the source of the HTTP connection, the hostname recorded in Apache access telemetry, the Fail2Ban detection and ban events, and connection tests demonstrating selective denial of service for an independently controlled legitimate IP. The manual result does not depend solely on an automated tool classification or on AI-generated interpretation.

What was assisted by artificial intelligence

Artificial-intelligence tools assisted with **reviewing publicly available source code and documentation** for Fail2Ban and other candidate applications, comparing implementation paths, organizing technical arguments, drafting English prose, designing diagrams and producing this report. AI-generated analysis, function references, interpretations and prose **may contain factual mistakes, omissions or outdated assumptions**.

Limits on reliance and generalization

- The reproduced production finding concerns a **specific configuration** of Apache and Fail2Ban. It is not a finding about every installation, released version, filter or firewall backend.
- Mentions of SSHGuard, CrowdSec, Wazuh, NGINX, pfSense or industrial platforms are **source-review candidates or contextual evidence**, not claims that those products were exploited in production.
- The payment-gateway and electric-power telemetry diagrams are **fictional impact scenarios**. No payment processing service, energy plant, railway network or physical process was attacked in this investigation.
- Readers should independently validate cited source paths, configuration defaults, security classifications and product-specific behavior before making technical, operational or legal decisions.

Responsibility for scientific interpretation remains human.

AI assistance can materially speed up research. It does not replace first-hand observation, reproducible testing, critical source verification or the accountability of the named author. The manually confirmed finding and the AI-assisted comparisons must not be conflated.

Author: Hugo Vázquez Caramés | **Organization:** Pentest®
Publication: 03 November 2026 | https://www.pentest.es